

# Design of Rate-Compatible Structured LDPC Codes for Hybrid ARQ Applications

Mostafa El-Khamy, Jilei Hou, and Naga Bhushan

**Abstract**—In this paper, families of rate-compatible protograph-based LDPC codes that are suitable for incremental-redundancy hybrid ARQ applications are constructed. A systematic technique to construct low-rate base codes from a higher rate code is presented. The base codes are designed to be robust against erasures while having a good performance on error channels. A progressive node puncturing algorithm is devised to construct a family of higher rate codes from the base code. The performance of this puncturing algorithm is compared to other puncturing schemes. Using the techniques in this paper, one can construct a rate-compatible family of codes with rates ranging from 0.1 to 0.9 that are within 1 dB from the channel capacity and have good error floors.

**Index Terms**—Low density parity check (LDPC) codes, hybrid automatic repeat request (HARQ), progressive-edge growth (PEG), progressive puncturing, rate compatible error correcting codes, protographs, channel coding for wireless applications.

## I. INTRODUCTION

**H**YBRID automatic repeat request (HARQ) mechanisms are deployed in wireless systems (e.g. 3GPP HSPA, LTE) to improve their capacity and peak rates. HARQ mechanisms can use forward error correcting codes to correct channel errors. Typically, in incremental redundancy HARQ mechanisms, the initial transmission consists of the data bits, encoded with an error detection code (e.g. CRC code) and a high rate forward error correcting code. If channel decoding of a transmission fails but a decoding error is detected, the receiver notifies the transmitter via a feedback channel and requests a retransmission via a negative acknowledgement (NAK) message. The subsequent retransmissions are not identical, but consists of parity bits which, when appended at the receiver to the bits from the previous (re)transmissions, form a code with a lower coding rate. Thus incremental redundancy HARQ mechanisms require rate-compatible families of codes. A rate-compatible family of codes suitable for HARQ applications is a nested family of codes with the same information block size but different coding rates. In such a rate-compatible family, the highest-rate code of the initial transmission should not only have a good performance alone, but should also have a good performance when the parity bits of the subsequent retransmissions are incrementally appended to it to form the lower-rate codes of the family.

Structured low density parity check (LDPC) codes have several advantages over non-structured LDPC codes due to their fast encoding and decoding structures and their natural scalability to different block sizes [1]–[3]. A structured LDPC code is characterized by a small graph called a *protograph*. A protograph  $\mathcal{G} = (\mathcal{V}, \mathcal{C}, \mathcal{E})$  is composed of a set of variable nodes  $\mathcal{V}$ , a set of check nodes  $\mathcal{C}$  and a set of edges  $\mathcal{E}$ , such that a variable node in  $\mathcal{V}$  can be connected to a check node in  $\mathcal{C}$  via one or more edges in  $\mathcal{E}$  [2]. The number of edges connected to a node is called the *degree* of the node. Each variable node, check node and edge in the protograph is said to be of a unique *type*. A *lifted graph* is constructed by making a number of copies of the protograph, determined by the desired information block size, and then carefully permuting the end-points of the edges between nodes of the same type in the protograph copies. It follows that the local neighborhood of any node in the lifted graph is the same as the local neighborhood of the corresponding node of the same type in the protograph. This facilitates faster optimization of structured LDPC codes by optimizing on the protograph level.

Rate-compatible structured LDPC codes were previously considered by Divsalar *et al.* [4]. The information block size of these families is not constant and thus they are not suitable for HARQ applications. A rate-compatible family of codes with constant information block size can be constructed from a low-rate base code where the higher-rate codes in the family are constructed from the base code by puncturing parity bits such that bits which are transmitted in one code are also transmitted in all codes of lower rate in the family. Rate-compatible families of LDPC codes designed by code expurgation and lengthening, as proposed by Dolinar [5], can also have a constant information block size. However, rate-compatible families with constant information block size constructed by puncturing are more convenient and have the advantage of using a single decoder for all the codes in the family, where the decoder simply assumes 50% reliability for the punctured bits. Recently Ha *et al.* proposed a puncturing algorithm for LDPC codes with short block lengths [6]. Based on asymptotic analysis, optimizing the fraction of variable nodes to be punctured from each set of variable nodes in the base code with a given node degree has been considered by Ha *et al.* [7]. However, this approach is not directly applicable to structured LDPC codes since it is often the case that all the parity variable nodes are of the same degree (i.e. degree 2). By modeling punctured nodes as erasures, it was recently shown in [8] that for LDPC codes there exists a cutoff rate  $R_c$  which depends on the degree distributions and the rate of the base code, such that through puncturing the base code, one could

Manuscript received 30 September 2008; revised 15 March 2009. The material in this paper was presented in part at the International Symposium on Information Theory, Seattle 2006.

The authors are with Corporate Research and Development, Qualcomm, San Diego (e-mail: mostafa@systems.caltech.edu, jhou@qualcomm.com and nbhush@qualcomm.com).

Digital Object Identifier 10.1109/JSAC.2009.090814.

not find a code that performs well under density evolution and whose rate is higher than  $R_c$ . Thus, to obtain a good family of rate-compatible codes by puncturing, it is essential to carefully choose the low-rate base code. Designing low-rate structured LDPC codes was considered by Divsalar *et al.* [9]. However, these codes have the property that the punctured (hidden) variable nodes have a very high degree and thus may not perform well under further puncturing.

A rate-compatible family of LDPC codes with constant information block size can also be constructed from a high-rate base code, where the lower-rate codes are constructed by extending the base code [10], [11]. It is also possible to construct lower-rate codes by information shortening [12]. However, the resulting codes in the family will not have a constant information block size and therefore will not be suitable for HARQ applications. Rateless (low rate) codes in the family can also be obtained by extending the base code through concatenating it with inner low density generator matrix (LDGM) codes, as suggested in [13], which is similar to Raptor codes [14].

In this paper, we design rate-compatible (RC) families of protograph-based LDPC codes for HARQ wireless applications. We have a two step approach to construct the rate-compatible families:

- I. Construct low-rate structured LDPC codes that can perform well as base codes for rate-compatible families with constant information block size. These low-rate base codes are optimized on the protograph level to have a good performance on both the additive white Gaussian noise channel and the binary erasure channel. A low-rate base code is then lifted from the optimized protograph to have the desired information block length, while maximizing its girth.
- II. Starting with the low-rate base code constructed in the previous step, construct a series of successively higher-rate codes, each of which is a punctured version of all lower-rate codes in the rate-compatible family. To ensure that the punctured codes have good performance, especially those with the highest coding rates in the family, the punctured parity bits are progressively selected using our proposed *progressive node puncturing (PNP) algorithm*. In the PNP algorithm, the nodes to be punctured from the last designed code are selected one at a time to maximize the likelihood that the message-passing algorithm will recover the punctured nodes.

This paper is organized as follows. In Section II, we consider the construction of good low-rate structured LDPC codes, that can serve as base codes for the rate-compatible family. These codes are designed to have low thresholds on the error channel as well as a good performance on the erasure channel. In Section III, we investigate how one can achieve a good family of rate-compatible codes by successively puncturing the low-rate base codes, designed in Section II, to obtain the higher-rate codes. We discuss puncturing algorithms for structured LDPC codes: regular puncturing, where puncturing of the parity bits follows a straightforward regular pattern, and random puncturing, where the the parity bits to be punctured are randomly chosen. We then propose a progressive node puncturing algorithm which progressively punctures the

variable nodes in the graph to obtain a higher-rate code that has a good performance with the message passing decoder. In Section IV, we discuss some examples of rate-compatible families constructed using the techniques described in this paper. In Section V, we draw some conclusions.

## II. DESIGN OF LOW-RATE BASE CODES

In this section, we design low-rate protograph codes that have a good performance on both the additive white Gaussian noise (AWGN) channel and the binary erasure channel in order to be robust against further puncturing. We devise a systematic technique to construct good low-rate protograph LDPC codes starting from good codes with a relatively higher rate. Compared to other low-rate structured LDPC codes, the hidden nodes in the codes designed here have a lower degree of connectivity.

### A. Construction Technique

It is desirable that LDPC codes have a large girth. A larger girth often corresponds to a better performance under message passing and a lower error floor in the high signal to noise ratio (SNR) region. A larger girth also corresponds to a larger stopping distance, (the stopping distance is the smallest number of erasures that cannot be recovered under iterative decoding) [15]. Base-codes with a larger stopping distance are expected to perform better under further puncturing. It is possible to construct the lifted graph of a base-code with a large girth if its corresponding protograph has a large girth (the lifted graph is loop-free if its protograph is loop-free).

Let  $\mathcal{B}$  be a base protograph, with  $N$  variable nodes,  $C$  check nodes, and  $K = N - C$  systematic input nodes. Assume that  $E$  of the redundancy variable nodes are punctured (hidden). The rate of the protograph  $\mathcal{B}$  is  $R_B = \frac{K}{N-E}$ . In the following algorithm, we will construct a protograph  $\mathcal{L}$  with rate  $R_L < R_B$  and girth as large as possible.

Let  $\mathcal{B}_\alpha$  be a lifted graph of  $\alpha$  copies of  $\mathcal{B}$ , then  $\mathcal{B}_\alpha$  is of the same rate as  $\mathcal{B}$  and has the same AWGN and erasure thresholds. As it will be shown below,  $\alpha$  is chosen to be a small number depending on both  $R_L$  and  $R_B$ . For example, if the base protograph is a rate-1/2 code,  $R_B = 1/2$ , and the target rate  $R_L = 1/T$ , where  $T$  is an integer greater than 2, then the desired number of copies of  $\mathcal{B}$  is  $\alpha = T - 1$ .

In the following algorithm, the lower-rate protograph  $\mathcal{L}$  is obtained from the protograph  $\mathcal{B}_\alpha$  by pruning nodes and edges from  $\mathcal{B}_\alpha$ . Since  $\mathcal{L}$  is a subgraph of  $\mathcal{B}_\alpha$ , it follows that the girth of  $\mathcal{L}$  is at least as large as the girth of  $\mathcal{B}_\alpha$ . Then maximizing the girth of  $\mathcal{B}_\alpha$  ensures that the girth of  $\mathcal{L}$  will be at least as large as this maximum. To maximize the girth of the lifted graph  $\mathcal{B}_\alpha$  with the desired length, it is constructed from the protograph  $\mathcal{B}$  by the progressive edge growth (PEG) algorithm [16] or its variations [9], [17]. Our codes are constructed by a version of the PEG algorithm adapted for protograph codes (described in the Appendix).

In our constructions, we will focus on systematic codes, i.e. codes generated with a systematic encoder. In a systematic protograph, the variable nodes can be classified as either input nodes or redundancy nodes. Systematic codes can be easily lifted from a systematic protograph. It is convenient to use

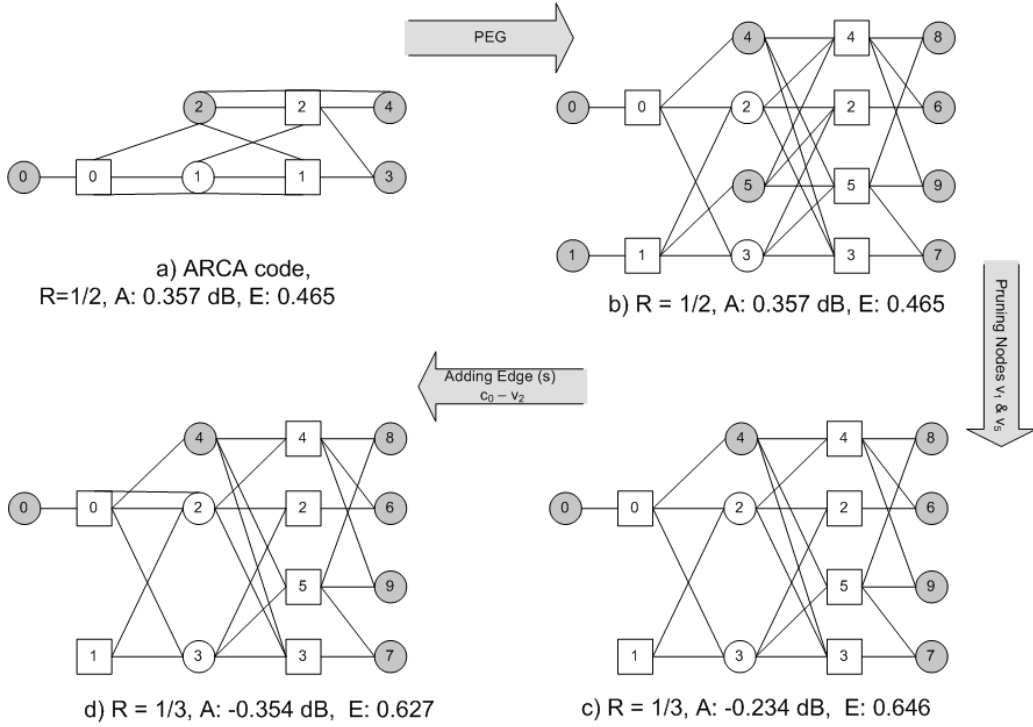


Fig. 1. Steps for construction of a protograph with rate 1/3 from the ARCA code with rate 1/2.

systematic codes in HARQ applications, where the initial transmission often consists of the systematic data bits and the data bits can be easily recovered in the case of no channel errors. Another advantage of using systematic codes is that a systematic low-rate code, constructed from a higher-rate code by pruning input nodes, can be conveniently generated using the same encoder as the higher-rate code.

The following algorithm gives a technique for deriving a protograph  $\mathcal{L}$ , of rate  $R_{\mathcal{L}} < R_{\mathcal{B}}$ , starting from a base protograph  $\mathcal{B}$ .

*Algorithm 1: Construction of low rate protographs*

- I. Copy the base graph,  $\mathcal{B}$ ,  $\alpha$  times.
- II. Construct another protograph  $\mathcal{B}_{\alpha}$  from the  $\alpha$  copies of  $\mathcal{B}$  using the PEG algorithm to maximize the girth of the protograph.
- III. Construct the protograph  $\mathcal{L}$  from  $\mathcal{B}_{\alpha}$  by pruning

$$\beta = \alpha K \frac{R_{\mathcal{B}} - R_{\mathcal{L}}}{R_{\mathcal{B}}(1 - R_{\mathcal{L}})} \quad (1)$$

input nodes and the edges connected to them.

- IV. The choice of the pruned nodes is optimized to get good AWGN and erasure thresholds.  $\square$

The PEG algorithm (or its variants) often has some randomness (as explained in the Appendix). In such a case, step II can be repeated to a maximum number of random trials and the protograph with the largest girth is chosen. In step IV, optimization can be done by choosing the pruned input nodes at random while using a gradient descent or simulated annealing approach [2] to pick the graph with the best threshold.

If the base protograph is a rate-1/2 code  $R_{\mathcal{B}} = 1/2$  and the target rate is  $R_{\mathcal{L}} = 1/T$ , where  $T$  is an integer greater than

2, then the design parameters can be

$$\alpha = T - 1 \quad \& \quad \beta = (T - 2)K. \quad (2)$$

For the special case when the number of pruned nodes,  $\beta$  is an integer multiple of  $K$ ,  $\beta = \gamma K$ , then  $\gamma$  copies of the  $K$  systematic input nodes can be pruned from  $\mathcal{B}_{\alpha}$ . Due to the symmetry of the protograph  $\mathcal{B}_{\alpha}$  across the  $\alpha$  copies, the choice of the pruned copies is unlikely to affect the performance. If no such symmetry exists, it is feasible to try all possible combinations of the  $\gamma$  copies and pick the combination that will give the lowest threshold. This is due to the small size of the protograph, which makes the search for a protograph threshold using density evolution (DE) [18] a fast process.

Additional optimization of the protograph can be done by adding, removing and swapping edges in  $\mathcal{L}$  without affecting the rate. In the following examples, we will demonstrate that carefully adding a very small number of edges is enough to get protographs with a better threshold. This step can also be automated in a simulated annealing setting. As will be shown in the construction examples below, improving the erasure threshold of the protograph may result in worsening its AWGN threshold. Since the protographs designed here will be used to construct base codes that will be punctured to construct RC families, more weight should be given to a better erasure threshold, especially if the design rate of the base code is very low.

Our approach to design a low-rate protograph from another medium-rate protograph can be considered information shortening: a medium-rate protograph is lifted by a small number of copies and then some information variable nodes and their connected edges are selectively pruned. The information shortening technique, proposed by Tian and Jones [12] for constructing low-rate LDPC codes from a medium-rate LDPC

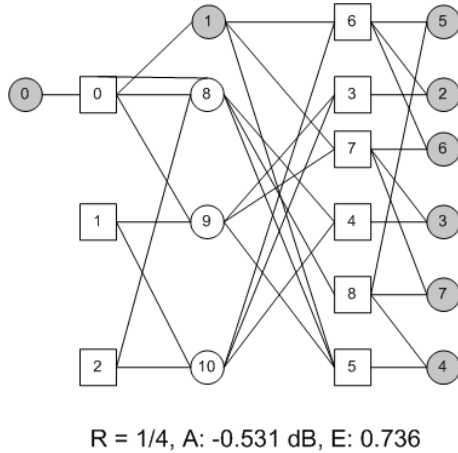


Fig. 2. Constructed protograph with rate  $1/4$ .

code, can also be applied here on the protograph level to design low-rate protographs. Following this technique, the degree distributions for the desired low-rate protograph can be optimized by Chung's Gaussian approximation [18] and fitted to the portion of the graph that will implement this rate.

### B. Construction Examples

In the figures accompanying the following examples, the AWGN and erasure thresholds are labeled by  $A$  and  $E$  respectively. Check and variable nodes will be represented by squares and circles respectively. Transmitted variable nodes are gray in color while punctured (non-transmitted) variable nodes are white. The rightmost variable nodes (of degree 2) are the transmitted redundant nodes. For asymptotic analysis (in the length of the code), we used density evolution to determine the AWGN and erasure thresholds [18].

The base code of choice is the rate- $1/2$  accumulate repeat check accumulate (ARCA) code, shown in Fig. 1.a, designed by Divsalar *et. al* [4]. The steps for constructing a protograph with rate  $1/3$  are shown in Fig. 1. The protograph in Fig. 1.b is a PEG-lifted graph of two copies of the ARCA code shown in Fig. 1.a. The rate- $1/3$  protograph in Fig. 1.c results by systematically pruning one copy of the input nodes. Its AWGN threshold is only 0.26 dB away from the AWGN capacity while its gap to the erasure channel capacity is 0.02. By simply adding one edge in Fig. 1.d, between  $c_0$  and  $v_2$  ( $c_i$  and  $v_i$  stand for the  $i$ th check and variable respectively) the gap to the AWGN capacity is reduced to only 0.14 dB while its gap to the erasure channel capacity is only 0.04. Both codes have a good performance on the erasure channel and the better AWGN threshold comes at a slight degradation in the performance on the erasure channel. In Fig. 2 and Fig. 3, we show protographs of rates  $1/4$  and  $1/5$  respectively, designed from the same ARCA base code of rate  $1/2$ . We also designed a protograph of a rate  $1/6$  that has an AWGN threshold of  $-0.7$  dB and an erasure threshold of 0.815. Further optimization of these constructed codes is still possible using simulated annealing.

We compared the performance of our structured LDPC codes with that of the RC 3GPP turbo codes for a payload of 4k (4096 bits) in Fig. 4. Our codes were constructed from the designed protographs using PEG. The number of half

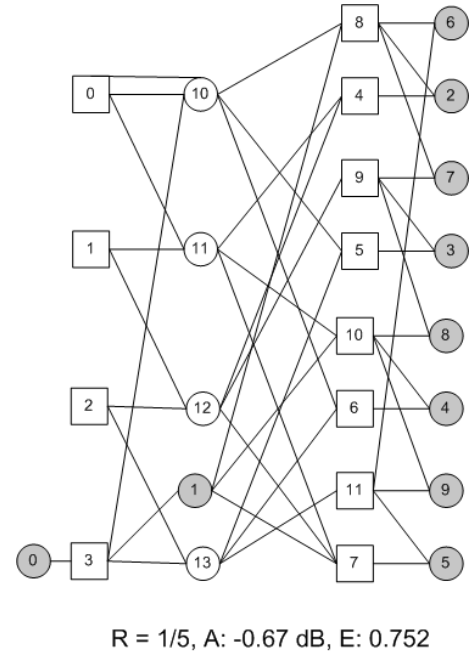


Fig. 3. Constructed protograph with rate  $1/5$ .

iterations for the turbo decoder is 30 and the maximum number of iterations for the LDPC belief propagation decoder is 192. In general, our codes have about 0.3 dB gain over turbo codes at a frame error rate (FER) of  $10^{-2}$  (which is the comparison criterion for mobile wireless applications) and have lower error floors than the turbo codes. It can also be observed that the error floors of the constructed structured LDPC codes seem to be getting better as their code rate decreases. This can be attributed to fewer cycles with small size in the Tanner graph of the codes with lower rate, resulting from pruning variable nodes and edges from the base protograph. One can also show that parallel decoding of structured LDPC codes, *cf.* [1], [3], can achieve much higher throughput (information bits decoded per second) than decoding of turbo codes with a parallel radix-4 turbo decoder.

## III. PUNCTURING ALGORITHMS

Previous techniques for constructing RC families and their suitability for HARQ applications using structured LDPC codes were discussed in the introduction (*cf.* [6] - [12]). Starting from a low-rate base code, we design a RC family of structured LDPC codes by sequential puncturing, where each code in the family can be obtained from a lower-rate code by puncturing a set of parity bits. We will focus on devising puncturing schemes that can result in a RC family of structured LDPC codes with a good performance across the desired rates.

### A. Structured Puncturing

Puncturing on the protograph level results in *structured* (regular) puncturing. This will have the advantage of low storage requirements. However, this approach has two major drawbacks. The first drawback is that the set of feasible rates attainable by structured puncturing is very small. Consider the ARCA code in Fig. 1.a. There are two redundancy symbols

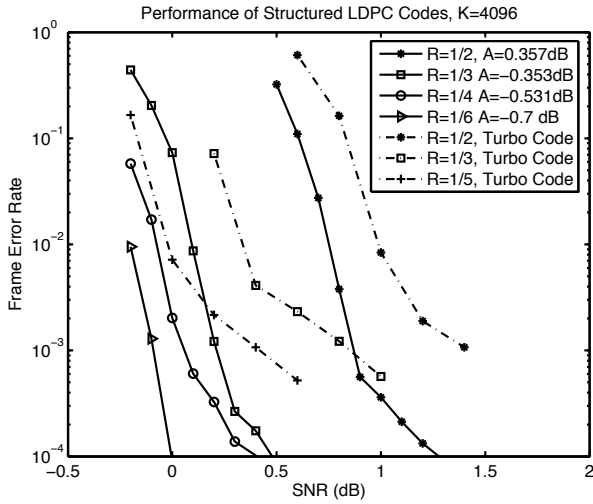


Fig. 4. Comparison of the constructed structured LDPC codes with 3GPP turbo codes ( $R$  denotes the rate of the code and  $A$  denotes the AWGN threshold of the code).

in the protograph. In this case, the feasible rates are only  $2/3$  (puncturing any of the two redundancy nodes) and unity (puncturing both redundancy nodes). The second major problem is that these structured protographs already have punctured (hidden) nodes, which are treated as erasures by the belief propagation (BP) algorithm. We observe that puncturing any of the two parity (redundancy) bits (*cf.* Fig. 1.a) results in all the checks in the protograph being connected to erased variables via at least two edges. In other words, the set of punctured nodes form a *stopping set* and iterative decoding can not start.

### B. Random Puncturing

Another straightforward approach is *random* puncturing. Random puncturing of the non-punctured redundancy variable nodes will appear as random noise on an erasure channel and the punctured nodes should be recovered if the code has a good performance on the erasure channel. The puncturing is such that variable nodes punctured in one code are also punctured in all codes of higher rate in the same family. We devised an algorithm that efficiently searches for the best random puncturing pattern which results in the code with the best AWGN threshold at each design rate: The search for the AWGN threshold is determined by running density evolution on the lifted graph. Let  $\mathcal{C}_R$  be a code constructed by random puncturing with rate  $R$  and threshold  $T$ . Let  $\mathcal{C}'_R$  be another code with rate  $R$  and obtained by another random puncturing pattern. If  $\mathcal{C}'_R$  achieves negligible error at an SNR  $T$ , then the AWGN threshold of  $\mathcal{C}'_R$  is found by bisecting the range of SNR less than  $T$ , and  $\mathcal{C}'_R$  becomes the best code at this rate, with an AWGN threshold  $T'$ . However, if  $\mathcal{C}'_R$  does not achieve negligible error at  $T$ , then  $\mathcal{C}'_R$  is discarded and  $\mathcal{C}_R$  remains the best code obtained by random puncturing at this rate. This process is repeated for a maximum number of random trials.

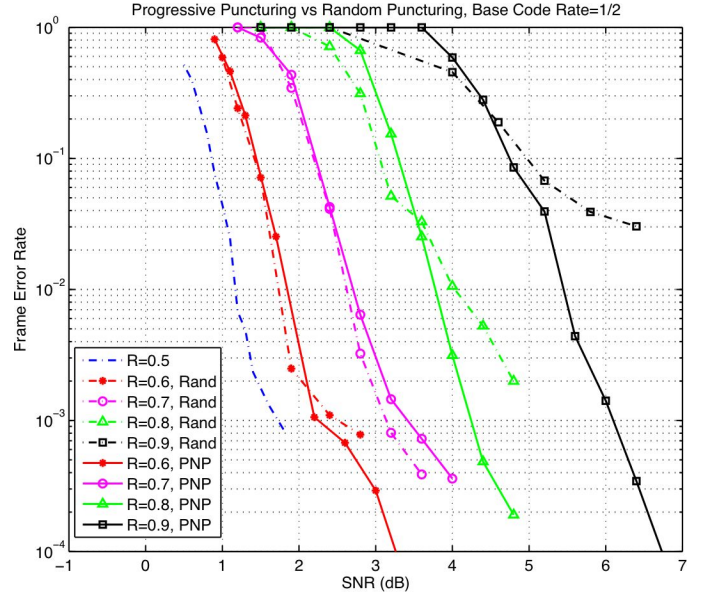


Fig. 5. FER of 1k Payload RC Families with progressive node puncturing (solid curves) and random puncturing (dotted curves). The base code is of rate  $R = 1/2$  and AWGN threshold 0.357 dB.

### C. Progressive Node Puncturing

Motivated by the previous discussion, we devise a systematic algorithm that progressively chooses the puncturing pattern that

- A. maximizes the number of checks in the graph which are directly connected to exactly one punctured variable node,
- B. minimizes the number of punctured variable nodes directly connected to the checks neighboring a punctured variable node, and
- C. maximizes the connectivity between punctured variable nodes and checks which are neighbors to at most one punctured variable node.

The reasoning behind criterion A is that a check (of degree  $> 1$ ) with only one punctured variable node connected to it will transmit non-zero information to the punctured node and the single punctured node could recover. However, maximizing the number of checks which have exactly one punctured neighbor, can result in some other checks being connected to a large number of punctured nodes. A check connected to more than one punctured node will initially transmit zero information to all the neighboring punctured nodes, until all but one of these punctured nodes are recovered later by message passing from other checks. Thus, it is crucial to also minimize the maximum number of punctured nodes connected to any check, which implies criterion B. To speed the recovery of the punctured nodes, the checks connected to at most one punctured variable should have a high connectivity to the other checks with more than one punctured neighbor, which is criterion C. In our algorithm, criterion A takes precedence over criterion B, which in turn takes precedence over criterion C.

Although maximizing the number of checks which are connected to exactly one punctured variable node may not result in the best code for this design rate, it ensures that

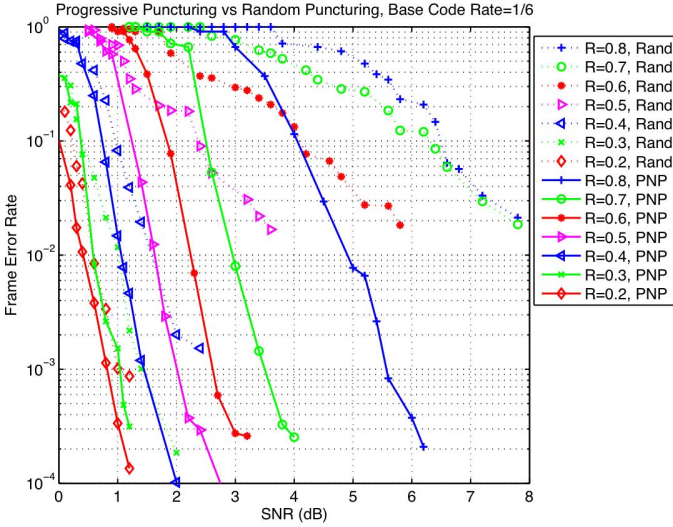


Fig. 6. FER of 1k payload RC families designed by random puncturing (dotted curves) and progressive node puncturing (solid curves). The base code is of rate  $R = 1/6$  and AWGN threshold  $-0.533$  dB.

this code will be robust for the further puncturing required to construct the next higher-rate code in the RC family. However, implementing criteria A and B in that order results in first (if possible) puncturing the variable node whose neighboring checks are each not currently connected to any other punctured variable node (ultimately rendering these checks to be connected to exactly one punctured variable node, and this node can be recovered with belief propagation decoding). This is then followed by puncturing (if possible) the variable node whose neighboring checks are each connected to more than one punctured variable node such that this variable node chosen to be punctured will directly affect the speed of recovery of the least number of already punctured variable nodes. In the proposed algorithm, the lifted code's variable nodes will be progressively punctured to satisfy criteria A and B. As will be shown below, the resulting puncturing pattern at a desired design rate can still be subject to some randomness. In this implementation, criterion C will not be assessed when puncturing the variable nodes one by one, but will rather be used to chose between candidate puncturing patterns for the desired design rate, where the graph with the largest *puncturing score* (cf. (5)) will be chosen.

Let  $\mathcal{C}(R_m)$  denote the code in the rate-compatible family with rate  $R_m$ . The numbers of variable nodes and check nodes in the Tanner graph of the lifted code are denoted by  $\tilde{N}$  and  $\tilde{C}$  respectively. Let  $\Psi_m$  be the set of non-punctured redundancy nodes in  $\mathcal{C}(R_{m-1})$  and  $\tilde{E}_{m-1}$  be the number of punctured variable nodes in the Tanner graph of  $\mathcal{C}(R_{m-1})$ . Let  $\mathcal{N}(x)$  denote the set of neighbors of the node  $x$  and  $|S|$  be the cardinality of the set  $S$ . In the following algorithm,  $\mathcal{C}(R_m)$  is obtained from  $\mathcal{C}(R_{m-1})$ , where  $R_m > R_{m-1}$ , by progressively puncturing  $P_m$  nodes from the set  $\Psi_m$  such that

$$P_m = \left\lfloor \tilde{N} - \frac{\tilde{N} - \tilde{C}}{R_m} - \tilde{E}_{m-1} \right\rfloor. \quad (3)$$

The following algorithm starts with a base code  $\mathcal{C}(R_0)$ , with rate  $R_0$ .

### Algorithm 2: Progressive Node Puncturing

Let  $\mathcal{C}(R_0)$  be the base code with rate  $R_0$ . Assume  $\mathcal{C}(R_{m-1})$  has been obtained from  $\mathcal{C}(R_0)$  by progressive node puncturing, then  $\mathcal{C}(R_m)$  is obtained by puncturing nodes from  $\mathcal{C}(R_{m-1})$  as follows:

- I. Find the set  $\Psi_m$  and calculate  $P_m$  by (3).
- II. For each check  $c \in \mathcal{N}(v)$  such that  $v \in \Psi_m$ , calculate
 
$$F(c) = |\{v : v \in \mathcal{N}(c) \text{ \& } v \text{ is punctured}\}|$$
- III. For each  $v \in \Psi_m$ , calculate
  - III. a.  $G(v) = |\{c : c \in \mathcal{N}(v) \text{ \& } F(c) = 1\}|$
  - III. b.  $H(v) = \sum_{c \in \mathcal{N}(v)} F(c)$
- IV. While  $p < P_m$  (puncture a node  $v_p$ )
  - IV. a. Find  $\Gamma = \{v_r : v_r \in \Psi_m \text{ \& } G(v_r) = \min_{v \in \Psi_m} G(v)\}$ .
  - IV. b. If  $|\Gamma| = 1$ ;  $v_p = \Gamma$ ;
  - IV. c. Else:
    - i) Find  $\Omega = \{v_r : v_r \in \Gamma \text{ \& } H(v_r) = \min_{v \in \Gamma} H(v)\}$ .
    - ii) If  $|\Omega| = 1$ ,  $v_p = \Omega$ .
    - iii) Else, choose  $v_p$  at random from  $\Omega$ .
  - IV. d. Puncture  $v_p$
  - IV. e. Update metrics
    - i) Remove  $v_p$  from  $\Psi_m$ :  $\Psi_m = \Psi_m \setminus v_p$ .
    - ii)  $\forall c \in \mathcal{N}(v_p)$ :  $F(c) := F(c) + 1$
    - iii)  $\forall v \in \mathcal{N}(c) \cap \Psi_m$  update  $H(v)$  and  $G(v)$ .
    - iv) Increment  $p$ ,  $p := p + 1$ .

The algorithm will start by puncturing the variable nodes whose neighboring checks are not connected to any punctured nodes ( $G(v) = 0$  &  $H(v) = 0$ , if they exist) since these variables will recover by iterative decoding and puncturing them will not introduce checks with more than one punctured neighbor. Proceeding to puncture the node  $v$  with the minimum  $G(v)$  and  $H(v)$  implements criteria A and B respectively. The algorithm can also be refined such that if there exist more than one variable with the same minimum  $G(v)$  and  $H(v)$  (Step IV.c.iii) then the variable in  $\Omega$  with the least degree of connectivity is punctured. If still more than one variable in  $\Omega$  have the same minimum degree, then one is chosen among them at random. This will also minimize the maximum number of punctured nodes connected to a check. To implement criterion C, we define the *puncturing score* of a check  $c$  to be the cardinality of the set of punctured variables connected to the checks reached by a two level expansion of the support tree of check  $c$ ,

$$\mathcal{S}(c) \triangleq \sum_{c' : c' \in \mathcal{N}(v) \text{ \& } v \in \mathcal{N}(c)} F(c'). \quad (4)$$

The puncturing score of the graph  $\mathcal{G}$  is defined by

$$\mathcal{S}_{\mathcal{G}} \triangleq \sum_{c \in \mathcal{G} \text{ \& } F(c) \leq 1} \mathcal{S}(c). \quad (5)$$

The puncturing score is a property of any Tanner graph with punctured variables. It serves as an approximate and efficient way to measure how well the checks with at most one punctured neighbor are connected to the other checks with punctured neighbors. It characterizes how fast the punctured



nodes can recover. At a slight increase in complexity, the progressive node puncturing (PNP) algorithm is modified to incorporate criterion C. For each design rate  $R_m$ , Algorithm 2 is repeated with a different random seed, for a small number of trials. One then chooses the puncturing pattern whose Tanner graph is associated with the largest puncturing score.

#### IV. CONSTRUCTION EXAMPLES OF RC FAMILIES

In this section we compare the performance of rate-compatible families constructed by progressive node puncturing and random puncturing. We consider rate-compatible families constructed directly from rate-1/2 base codes as well as families that are constructed from low-rate base codes and cover a wider range of code rates.

First, we construct RC families using the rate-1/2 ARCA code as the base code. The RC families are constructed for payloads (number of systematic bits) of 1k and 4k using (greedy) random puncturing and progressive node puncturing. Fig. 5 shows the performance of the 1k payload families on the AWGN channel (for the sake of figure clarity only codes with rate increments of 0.1 are shown). The codes designed by PNP have lower error floors at the higher rates (0.8 and 0.9). This exemplifies the crux behind the PNP algorithm, where instead of constructing the best punctured code for rate 0.7, the PNP constructs a good punctured code with rate 0.7, that when subjected to further puncturing will result in good higher-rate (0.8 and 0.9) codes. This is important since the error performance of the highest rate code of the initial HARQ transmission, plays an important role in the efficiency of the HARQ process, since subsequent transmissions occur only if the initial transmission is in error (*cf.* [10]).

In the following example, we will show that the advantage of the PNP algorithm over other puncturing algorithms is amplified when the base code has a low rate. Using the techniques in Section II, we designed two codes of rate 1/6:  $C_1$  which has an AWGN threshold of  $-0.54$  dB and has a 0.01 gap to the erasure channel capacity and  $C_2$  which has an AWGN threshold of  $-0.7$  dB and has a 0.019 gap to the erasure channel capacity (the performance of  $C_2$  for a 4k payload is shown in Fig. 4). From the base codes of rate 1/6, we constructed RC codes whose rates range from 0.2 to 0.8 with increments of 0.05. Although  $C_2$  has a better AWGN threshold than  $C_1$ , the RC families designed with  $C_2$  as a base code had a worse performance due to its poorer erasure threshold. The performance of the RC family for a 1k payload, with a base code  $C_1$  of rate 1/6, designed by progressive node puncturing is compared to that obtained by random puncturing in Fig. 6 (for figure clarity, only codes with rate increments of 0.1 are shown). The code of rate 0.8 in the RC family constructed from  $C_1$  by PNP is 0.86 dB away from the capacity while the code of rate 0.8 in the RC family constructed by random puncturing is 2.4 dB away from the capacity. Fig. 6 shows that the RC family obtained by PNP has a superior frame error rate (FER) performance over that obtained by random puncturing, especially for rates greater than 1/2 when random puncturing results in severe error floors.

In Fig. 7, the performance of two RC families with 1k payload and obtained by progressive node puncturing is shown.

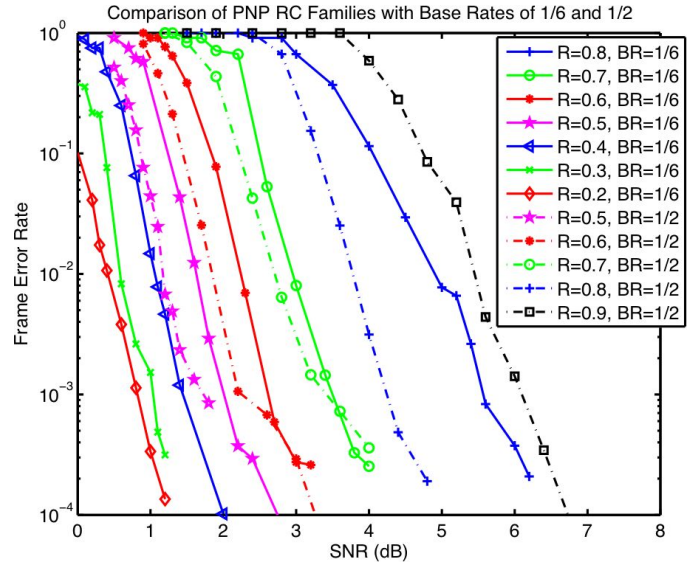


Fig. 7. Comparison of frame error rate of two 1k payload RC families designed by progressive node puncturing from base codes with base rates BR=1/6 and BR=1/2 respectively.

The base code of the first family is  $C_1$  whose AWGN threshold is  $-0.54$  dB and rate is 1/6. The base code of the second family is  $C_0$ , the rate-1/2 ARCA code with AWGN threshold 0.357 dB. It is observed that the rate-1/2 code obtained from  $C_1$  by PNP, has a slightly worse performance, 0.5 dB penalty at  $10^{-2}$  FER, than the rate-1/2 ARCA base-code  $C_0$ . Similarly, codes with rates 0.6, 0.7 and 0.8 obtained by PNP from  $C_1$  will perform slightly worse than these codes with the same rates obtained by PNP from  $C_0$ . This is expected since the rate-1/2 code constructed by puncturing  $C_1$  has a larger fraction of its variable nodes punctured (or hidden) compared to  $C_0$  (13/17 punctured and hidden bits for the former compared to 1/5 for  $C_0$ ). At high SNRs, the throughput and delay performance of HARQ schemes is dominated by the error rate of the initial transmission (i.e. the selected highest-rate code), whereas at low SNRs, e.g. due to fading, the packet failure probability will be dominated by the performance of the lowest-rate code in the RC family. At a target FER of  $10^{-2}$ , the RC family with base-code rate of 1/6 provides reliable performance at an SNR as low as 0.4 dB whereas the RC family with base-code rate of 1/2 cannot provide reliable performance at an SNR lower than 1.2 dB. Thus RC families constructed from base codes with low coding rates are more robust to a wide range of operating conditions. The code rates selected for the initial transmission and the retransmissions, and the number of retransmissions can be optimized according to the target FER, desired latency and throughput of the system.

#### V. CONCLUSIONS

We considered the design of rate-compatible LDPC code families for hybrid ARQ applications: First, we devised low complexity systematic techniques for designing low-rate structured LDPC codes from higher-rate ones. The codes have a good performance on both the erasure and AWGN channels and are expected to perform well under further puncturing. These codes will serve as the base codes in a RC family.

The structure of these codes makes them suitable for high throughput applications due to the inherent parallelization of the decoder. Second we devised puncturing algorithms to obtain higher rate codes, in the RC family, from the base code. We point out here some general observations on the puncturing algorithms. *Structured puncturing* (puncturing on the protograph level) will not in general work as it will more likely introduce stopping sets. *Random puncturing search* greedily finds the best (random) puncturing pattern for each successive code in the family. Thus, the low rate codes in the family may perform well while the higher rate codes will suffer from error floors. *Progressive node puncturing* carefully assigns the puncturing pattern such that further puncturing of this code would result in good higher rate codes. The designed codes will have good thresholds over the whole family range as well as good error floors.

#### ACKNOWLEDGMENT

The authors are grateful to Henry Pfister for providing a program for the progressive edge growth (PEG) algorithm. The authors are also very grateful to the editors of this special issue and the reviewers for their comments which improved the quality and presentation of this paper.

#### APPENDIX: THE PROGRESSIVE EDGE GROWTH ALGORITHM

In this appendix, we briefly describe a variation of the progressive edge growth (PEG) algorithm, by Hu *et al.* [16], used in our constructions. The progressive edge growth (PEG) algorithm is a greedy algorithm that constructs LDPC codes on an edge by edge basis such that it maximizes the local girth at each variable node. The PEG algorithm was originally proposed for regular and irregular (unstructured) ensembles of LDPC codes. It assumes the knowledge of the number of check nodes, the number of variable nodes and the degree distributions of the variable nodes in the final graph. The following algorithm is used to construct structured LDPC codes from a protograph, where each edge, variable node and check node is said to have a unique type.

*Algorithm 3:* (Modified) PEG Algorithm:

- I. For each edge type
  - I. a. For each bit, in the lifted graph, valid for that edge type
    - i) Find a list of valid checks
    - ii) Construct the support tree for the considered bit till the maximum depth while not all valid checks are in the tree.
    - iii) Construct a list of valid checks not in the support tree.
    - iv) Pick one of these checks at random and connect it to the considered bit via the considered edge.
    - v) Mark the connected check and bit as invalid for future connections to the same edge type.

This algorithm progressively adds edges in the lifted graph, where edges of the same type are added sequentially. According to its type, an edge could only connect between checks and variables of valid types as dictated by the protograph. The

algorithm could be repeated a number of times each with a different random seed (*cf.* step I.a.iv). Then the lifted code with the smallest number of cycles of smallest size is chosen.

#### REFERENCES

- [1] T. Richardson and V. Novichkov, "Methods and apparatus for decoding LDPC codes," *United States patent 6,633,856*, Oct. 2003.
- [2] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progress Report 42-154, JPL*, Aug. 2003.
- [3] J.-S. Lee, B. Lee, J. Thorpe, K. Andrews, S. Dolinar, and J. Hamkins, "A scalable architecture of a structured LDPC decoder," in *International Symposium on Information Theory, 2004. ISIT 2004*, June 2004, p. 292.
- [4] D. Divsalar, S. Dolinar, J. Thorpe, and C. Jones, "Constructing LDPC codes from simple loop-free encoding modules," in *2005 IEEE International Conference on Communications*, pp. 658 – 662.
- [5] S. Dolinar, "A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening," in *Proc. ISIT*, 2005, pp. 1622–1626.
- [6] J. Ha, J. Kim, D. Klinc, and S. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inform. Theory*, vol. 52, pp. 728–738, Feb 2006.
- [7] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.
- [8] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes," in *2004 IEEE Information Theory Workshop, San Antonio, Texas*, Oct. 2004.
- [9] D. Divsalar, S. Dolinar, and C. Jones, "Low-rate LDPC codes with simple protograph structure," in *Proc. ISIT*, 2005, pp. 1627–1631.
- [10] J. Li and K. R. Narayanan, "Rate-compatible low density parity check (RC-LDPC) codes for capacity-approaching ARQ schemes in packet data communications," in *Proceeding of International Conference on Communications, Internet and Information Technology (CIIT), US Virgin Islands*, Nov. 2002, pp. 201–206.
- [11] M. Yazdani and A. Banihashemi, "On construction of rate-compatible low-density parity-check codes," *IEEE Commun. Lett.*, vol. 8, pp. 159–161, Mar. 2004.
- [12] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP JWCN 2005:5 (2005)*, pp. 789–795.
- [13] T. J. Richardson and R. Urbanke, "Multi-edge type LDPC codes," preprint. [Online]. Available: <http://lthcwwww.ep.ch/papers/multiedge.ps>.
- [14] A. Shokrollahi, "Raptor codes," in *Proceedings 2004 IEEE International Symposium on Information Theory (ISIT2004), Chicago, IL, USA*, 2004.
- [15] A. Orlitsky, R. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and the girth of Tanner graphs," in *2002 IEEE International Symposium on Information Theory*, 2002.
- [16] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, pp. 386 – 398, Jan. 2005.
- [17] H. Xiao and A. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, pp. 715 – 717, Dec. 2004.
- [18] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, pp. 657–670, 2001.





**Mostafa El-Khamy** received the Ph.D. degree in electrical engineering in 2006 from the California Institute of Technology (Caltech), Pasadena, USA where he also received an M.S. degree in electrical engineering in 2003. He received the B.S. degree (with honor) and an M.S. degree in electrical engineering (communications and electronics section) from Alexandria University, Egypt in 1999 and 2001 respectively. He has been a senior systems engineer in Corporate Research and Development at Qualcomm, San Diego, USA from October 2006

to October 2008 where he has contributed to the design and development of next generation mobile wireless systems, with an emphasis on 3GPP WCDMA/HSPA+ evolution. He has also been a key member in the analysis and design of advanced solutions for the deployment of femto-cells and home access points. He is currently an assistant professor at Alexandria University, Egypt and a visiting lecturer at the German University in Cairo, Egypt. He received the Atwood Fellowship from the California Institute of Technology in 2002. He is also the recipient of the Young Scientist Award from the URSI General Assembly, the Netherlands, in 2002. His research interests include wireless communications, information theory, coding theory and signal processing.



**Jilei Hou** (S00-M04) received the B.E. degree (with highest honor) and the M.E. degree in Radio Engineering from Southeast University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical Engineering from the University of California, San Diego, La Jolla, CA, in 2003. He is currently a Senior Staff Engineer and Manager in the Corporate Research and Development Division at QUALCOMM, Inc. He joined QUALCOMM in May 2003, where he has been a key member involved in the performance analysis, system design,

and development of advanced techniques for CDMA2000 EV-DO systems, including interference cancellation and MIMO. He currently leads a team on system algorithm and performance enhancements for WCDMA/HSPA+ evolution. He has published over 20 papers in international conferences and journals and filed about 20 patents in the area of wireless communications.



**Naga Bhushan** obtained his B. Tech. degree in Electronics from I.I.T. Chennai, India in 1989. He pursued his graduate study at Cornell University, where he completed his M. S. and Ph. D. in 1992 and 1994 respectively, both in Electrical Engineering. He has been working as a Systems Engineer at Qualcomm since 1994, where he is now a Vice President of Technology in the Corporate R&D group. During the course of his work at Qualcomm, Naga Bhushan has been involved in the design and development of advanced wireless communication systems, with

emphasis on channel coding, link adaptation techniques, modem design and performance optimization for CDMA and OFDMA-based high-speed wireless packet data systems, including cdma2000 EV-DO, Ultra Mobile Broadband (UMB) and 3GPP LTE-Advanced.